

The application is amended as follows:

- Claims 1, 10, 20-23, and 25 are amended; and
- Claims 15 and 27-28 are cancelled from the application

A markup of the Claims is provided on sheets 2-6; Remarks begin at sheet 9.

AMENDMENTS TO THE CLAIMS

1. (Currently amended) In a directory server containing heterogenous directory entries, a method of hierarchically navigating said entries comprising the steps of:
 - creating one or more directory views;
 - organizing said directory views into a hierarchy using only information concerning said entries; and
 - using one of said directory views that is most appropriate for navigating to said entries.
2. (Original) The method of Claim 1, wherein said entries do not need to be physically in any particular place.
3. (Original) The method of Claim 1, wherein said directory server may have a flat directory information tree.
4. (Original) The method of Claim 1, wherein the existence of said directory views is transparent to a client of said directory server and said client is not required to have special knowledge of said directory views to use them.

5. (Original) The method of Claim 1, wherein each of said directory views begins with an ordinary entry.

6. (Original) The method of Claim 1, wherein each of said directory views belongs to a specific object class that contains a filter attribute, said filter attribute containing a filter that describes said views.

7. (Original) The method of Claim 6, wherein said filter attribute is omitted from said views to facilitate a hierarchical directory structure.

8. (Original) The method of Claim 1, wherein each of said directory views comprises sub-views which provide a subset of said views.

9. (Original) The method of Claim 8, wherein said sub-views comprise different subject domains from said directory views.

10. (Currently amended) In a directory server containing heterogeneous directory entries and a directory views hierarchy, each view containing a filter describing said view, a method of searching said view in said directory views hierarchy with a given filter, comprising the steps of:

rewriting said given filter to be a sub-tree search of the parent of the topmost view in said view hierarchy; and

performing said sub-tree search with said rewritten filter;

wherein each of said directory views belongs to a specific object class that contains a filter attribute, said filter attribute containing a filter that describes said directory views.

11. (Original) The method of Claim 10, wherein said directory entries do not need to be physically in any particular place.

12. (Original) The method of Claim 10, wherein said directory server has a flat directory information tree.

13. (Original) The method of Claim 10, wherein the existence of said views is transparent to a client of said directory server and said client requires no special knowledge of said views to use them.

14. (Original) The method of Claim 10, wherein each of said directory views begins with an ordinary entry.

~~15. (Original) The method of Claim 10, wherein each of said directory views belongs to a specific object class that contains a filter attribute, said filter attribute containing a filter that describes said directory views.~~

16. (Original) The method of Claim 10, wherein said filter attribute is omitted from said directory views to facilitate a hierarchical directory structure.

17. (Original) The method of Claim 10, wherein each of said directory views comprises sub-views which provide a subset of said views.

18. (Original) The method of Claim 17, wherein said sub-views comprise different subject domains from said directory views.

19. (Original) The method of Claim 10, said rewriting step comprising:

(a) collecting filters from said view and all ancestor views of said view to form a first sub-filter;

(b) if the search is not a sub-tree search, collecting all filters from all descendent views to form a second sub-filter;

(c) adding a third sub-filter to ensure all children of said view are included in the search for one level search or ensure all descendents of said view are included for a sub-tree search; and

(d) combining said sub-filters from steps (a)-(c) and said given filter to produce said rewritten filter.

20. (Currently amended) The method of Claim 19, wherein step (a) further comprises the steps of:

(1) starting from the topmost view and working down;

(2) adding each filter to said first sub-filter in step (a) using the logical AND operator; and

(3) moving down said hierarchy and going to step (2) until at said view.

21. (Currently amended) The method of Claim 19, wherein step (b) further comprises the steps of:

- (1) working down said hierarchy until said hierarchy ends;
- (2) adding each filter to said second sub-filter in step (b) using the logical AND operator and the logical NOT operator;
- (3) repeating step (2) until all sub-views of said view have been accounted for.

22. (Currently amended) The method of Claim 19, wherein step (c) further comprises the steps of:

- (1) for sub-tree searches, using the logical OR operator and a filter which includes the components of said descendent views' distinctive attributes, and which excludes the distinctive attribute of said view;
- (2) for one level searches, using the logical OR operator and a filter which includes the components of said children views' distinctive attributes, and which excludes the relative distinctive attribute of all children views of said view using the logical NOT operator; and
- (3) for base searches, using the filter "objectclass=nsview", wherein "nsview" is the object class of said views.

23. (Currently amended) The method of Claim 19, wherein step (d) further comprises the steps of:

- (1) combining said third sub-filter from step (c) with the given search filter using the logical AND operator;

(2) combining said first sub-filter from step (a) and said second sub-filter from step (b) with the given search filter using the logical AND operator; and

(3) combining the resulting filters from step (1) and (2) using the logical operator OR.

24. (Original) The method of Claim 19, wherein said sub-filters from steps (a), (b) and (c) may be cached so that the filter rewriting only needs to perform step (d), which amounts to simple filter concatenation.

25. (Currently amended) A directory server for managing heterogeneous directory information, comprising:

a plurality of directory entries; and

a flat directory information tree;

a set of directory views to facilitate hierarchical navigation of said directory entries; and

means to search said directory views by rewriting filters.

26. (Original) The directory server of Claim 25, wherein said directory entries do not need to be physically in any particular place.

~~27. (Original) The directory server of Claim 25, further comprising:~~

~~a flat directory information tree.~~

~~28. (Original) The directory server of Claim 27, further comprising:~~

~~means to search said directory views by rewriting filters.~~

29. (Original) The directory server of Claim 25, wherein the existence of said directory views is transparent to a client of said directory server and said client is not required to have special knowledge of said directory views to use them.

30. (Original) The directory server of Claim 25, wherein each of said directory views begins with an ordinary entry.

31. (Original) The directory server of Claim 25, wherein each of said directory views belongs to a specific object class that contains a filter attribute, said filter attribute containing a filter that describes said directory views.

32. (Original) The directory server of Claim 25, wherein said filter attribute is omitted from said views to facilitate a hierarchical directory structure.

33. (Original) The directory server of Claim 25, wherein each of said directory views comprises sub-views that provide a subset of said directory views.

34. (Original) The directory server of Claim 33, wherein said sub-views comprise different subject domains from said directory views.